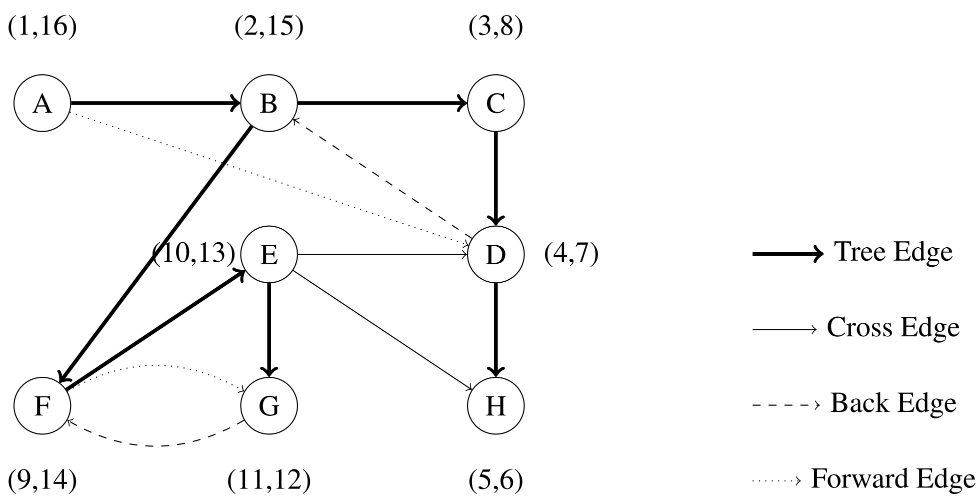
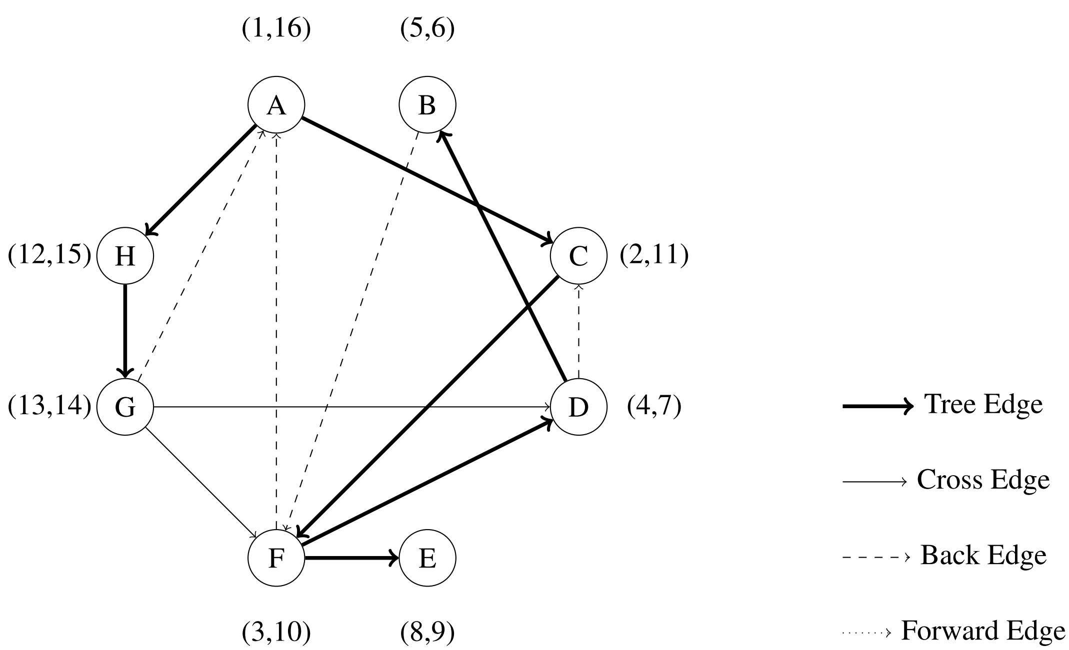
1. (21 pts.) Problem 1

(a)



(b)



2. (20 pts.) Problem 2

(a) $A : 1, 14$
$B : 15, 16$
$C : 2, 13$
$D : 3, 10$
$E : 11, 12$
$F : 4, 9$
$G : 5, 6$
$H : 7, 8$

(b) Sources: A, B; Sinks: G, H

(c) B, A, C, E, D, F, H, G

(d) Any ordering of the graph must be of the form $\{A, B\}, C, \{D, E\}, F, \{G, H\}$, where $\{A, B\}$ indicates A and B may be in any order within these two places. So, the total number of orderings is $2^3 = 8$.

3. (20 pts.) Problem 3

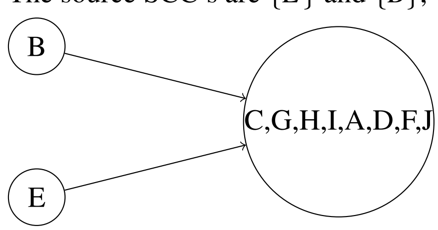
We can't just use DFS, since that takes time $O(|V| + |E|)$. However, we can modify DFS to terminate the first time we see an edge that goes back to a visited vertex; this a minor modification to the Explore procedure. The algorithm will detect if there is a cycle.

Let us analyze the running time of this modified DFS algorithm. The key observation is that a graph with more than $|V|$ edges will always have a cycle. (If you don't see why, you should prove this fact. Essentially, if $|E| \geq |V|$ your are force to create a cycle.) Suppose first that the graph has no cycles. Then, $|E| \leq |V| - 1$. So, in this case, the running time of $O(|E| + |V|)$ of DFS is actually $O(|V|)$. Now, suppose that the algorithm stopped early. This is because it found some edge coming from the currently considered vertex to a vertex that has already been considered. Since all of the edges considered up to this point didn't do that, we know that they formed a forest. So, the number of edges considered is at most the number of vertices considered, which is $O(|V|)$. Consequently, the total running time is $O(|V|)$.

4. (20 pts.) Problem 4

(i) (a) The strongly connected components are found in the order $\{C, G, H, I, A, D, F, J\}$, $\{E\}$, $\{B\}$.

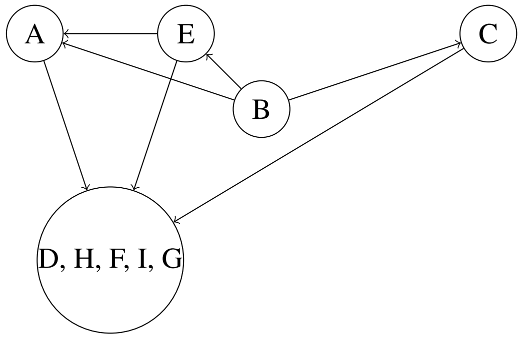
(b) The source SCC's are $\{E\}$ and $\{B\}$, while $\{C, G, H, I, A, D, F, J\}$ is a sink SCC.

(c) 

(d) It is necessary to add two edges to make the graph strongly connected, e.g. by adding $C \to B$ and $B \to E$.

(ii) (a) The strongly connected components are found in the order $\{D, F, G, H, I\}$, $\{C\}$, $\{A\}$,{E}, $\{B\}$.

(b) $\{B\}$ is a source SCC, while $\{D, F, G, H, I\}$ is a sink.

(c) 

(d) In this case, adding one edge from any vertex in the sink SCC to any vertex in the source SCC makes the metagraph strongly connected and hence the given graph also becomes strongly connected.

5. (19 pts.) Problem 5

(a) Let $G = (V, E)$ be our (directed) graph. The set of nodes in our graph shall be the integers modulo 600, representing values of bills the machine might print: $V = \{0, 1, 2, \ldots, 599\}$. An edge between two nodes v_0 and v_1 exists iff the machine is willing to print bills of value v_1 when given bills of value v_0. The question: is there a path from node 1 to node 10?

(b) Given the above description, it is easy to see that a Explore algorithm on that graph should be applied, starting from node 1, with an extra line of code that halts and answers "YES" if node 10 is reached, and "NO" if all the connected component of the starting node is exhausted and node 10 is not reached.

6. (0 pts.) Acknowledgments

(a) I did not work in a group.
(b) I did not consult with anyone other than my group members.
(c) I did not consult any non-class materials.

Rubric:

Problem 1, 21 pts

(a)
- 0.5 pts for every node's pre number and post number
- 0.5 pts for every edge type

(b)
- 0.5 pts for every node's pre number and post number
- 0.5 pts for every edge type

Problem 2, 20 pts

(a) 8 points: 0.5 point for a correct pre number and a correct post number, respectively, for each vertex.

(b) 2 points: 0.5 point for each correctly categorized vertex.

(c) 4 points: 0.5 point for each correctly ordered vertex.

(d) 6 points
3 points for the correct answer
3 points for a reasonable explanation

Problem 3, 20 pts

(a) 5 pts: Relating the back edge found by DFS to existence of a cycle in undirected graph.

(b) 10 pts: modifying the Explore procedure such that it finds a back edge, and terminates early (just an explanation would be enough, they do not need to write a pseudocode)

(c) 5 pts: analyzing the runtime of the algorithm, and explain why it would be independent of $|E|$ if we terminate the algorithm early

Problem 4, 20 pts

(a) 2 points for (i) and (ii), respectively

(b) 2 points for (i) and (ii), respectively

(c) 4 points for (i) and (ii), respectively

(d) 2 points for (i) and (ii), respectively

Problem 5, 19 pts

(a)
- 2 pts for directed graph
- 2 pts for the right description of node
- 2 pts for the right description of edge
- 3 pts for the questions raised by this problem

(b)
- 5 pts for DFS/Explore
- 2.5 pts for the conclusion of ”YES”
- 2.5 pts for the conclusion of ”NO”